

```

//File baseTypes.hh

#ifndef NULL
#define NULL (0)
#endif

typedef unsigned short UShort;
typedef short Short;
typedef unsigned long ULONG;
typedef long Long;
typedef double Double;
typedef float Float;

enum Boolean {
    RTI_FALSE = 0,
    RTI_TRUE};

class Exception {
public:
    RTI::ULONG _serial;
    char *_reason;
    const char *_name;
    Exception (const char *reason);
    Exception (RTI::ULONG serial, const char *reason=NULL);
    Exception (const Exception &toCopy);
    virtual ~Exception ();
    Exception & operator = (const Exception &);
    friend ostream& operator<< (ostream &, Exception *);
};

#define RTI_EXCEPT(A) \
class A : public RTI::Exception {      \
public: \
    static const char *_ex; \
    A (const char *reason) : Exception (reason) { _name = _ex; \
} \
    A (RTI::ULONG serial, const char *reason=NULL) \
        : Exception (serial, reason) { _name = _ex; } \
    A (const RTI::Exception &toCopy) : Exception(toCopy) { \
        _name = _ex; } \
};


```

```

//File: RTItypes.hh

// RTI Parameter Passing Memory Conventions
// C1 In parameter by value.
// C2 Out parameter by reference.
// C3 Function return by value.
// C4 In parameter by const reference. Caller provides memory.
// Caller may free memory or overwrite it upon completion of
// the call. Callee must copy during the call anything it
// wishes to save beyond completion of the call. Parameter
// type must define const accessor methods.
// C5 Out parameter by reference. Caller provides reference to object.
// Callee constructs an instance on the heap (new) and returns.
// The caller destroys the instance (delete) at its leisure.
// C6 Function return by reference. Callee constructs an instance on
// the heap (new) and returns a reference. The caller destroys the
// instance (delete) at its leisure.

#define MAX_FEDERATION          20
#define MAX_FEDERATE             20
#define MAX_FEDERATION_NAME_LENGTH 64
#define MAX_FEDERATE_NAME_LENGTH 64
#define MAX_OBJECT_CLASSES        100
#define MAX_INTERACTION_CLASSES   100
#define MAX_ATTRIBUTES_PER_CLASS 50
#define MAX_PARAMETERS_PER_CLASS 50
#define MAX_BYTES_PER_VALUE       512
#define PRIVILEGE_TO_DELETE_HANDLE 0
#define ROOT_OBJECT_CLASS_HANDLE   0
#define ROOT_INTERACTION_CLASS_HANDLE 0
#define ROOT_OBJECT_CLASS_NAME     "root"
#define ROOT_INTERACTION_CLASS_NAME "root"
#define PRIVILEGE_TO_DELETE_NAME   "privilegeToDelete"
#define MAX_USER_TAG_LENGTH        16
#define RTI_VERSION                "1.0R2"

#define EPSILON (1.0e-9)
//as defined for IEEE 754
//see <math.h>
#define POSITIVE_INFINITY (HUGE_VAL)

RTI_EXCEPT(ArrayIndexOutOfBoundsException)
RTI_EXCEPT(AttributeAlreadyOwned)
RTI_EXCEPT(AttributeNotDefined)
RTI_EXCEPT(AttributeNotKnown)
RTI_EXCEPT(AttributeNotOwned)
RTI_EXCEPT(AttributeNotPublished)
RTI_EXCEPT(AttributeNotSubscribed)
RTI_EXCEPT(ConcurrentAccessAttempted)
RTI_EXCEPT(CouldNotDiscover)
RTI_EXCEPT(CouldNotOpenFED)
RTI_EXCEPT(CouldNotRestore)
RTI_EXCEPT(DeletePrivilegeNotHeld)
RTI_EXCEPT(ErrorReadingFED)
RTI_EXCEPT(EventNotKnown)
RTI_EXCEPT(FederateAlreadyPaused)
RTI_EXCEPT(FederateAlreadyExecutionMember)
RTI_EXCEPT(FederateDoesNotExist)
RTI_EXCEPT(FederateInternalError)
RTI_EXCEPT(FederateNameAlreadyInUse)
RTI_EXCEPT(FederateNotExecutionMember)
RTI_EXCEPT(FederateNotPaused)

```

```

RTI_EXCEPT(FederateOwnsAttributes)
RTI_EXCEPT(FederatesCurrentlyJoined)
RTI_EXCEPT(FederationAlreadyPaused)
RTI_EXCEPT(FederationExecutionAlreadyExists)
RTI_EXCEPT(FederationExecutionDoesNotExist)
RTI_EXCEPT(FederationNotPaused)
RTI_EXCEPT(FederationTimeAlreadyPassed)
RTI_EXCEPT(HandleValuePairMaximumExceeded)
RTI_EXCEPT(IDsupplyExhausted)
RTI_EXCEPT(InteractionClassNotDefined)
RTI_EXCEPT(InteractionClassNotKnown)
RTI_EXCEPT(InteractionClassNotPublished)
RTI_EXCEPT(InteractionParameterNotDefined)
RTI_EXCEPT(InteractionParameterNotKnown)
RTI_EXCEPT(InvalidDivestitureCondition)
RTI_EXCEPT(InvalidExtents)
RTI_EXCEPT(InvalidFederationTime)
RTI_EXCEPT(InvalidLookahead)
RTI_EXCEPT(InvalidObjectID)
RTI_EXCEPT(InvalidOrderType)
RTI_EXCEPT(InvalidResignAction)
RTI_EXCEPT(InvalidRetractionHandle)
RTI_EXCEPT(InvalidTransportType)
RTI_EXCEPT(MemoryExhausted)
RTI_EXCEPT(NameNotFound)
RTI_EXCEPT(NoPauseRequested)
RTI_EXCEPT(NoResumeRequested)
RTI_EXCEPT(ObjectClassNotDefined)
RTI_EXCEPT(ObjectClassNotKnown)
RTI_EXCEPT(ObjectClassNotPublished)
RTI_EXCEPT(ObjectClassNotSubscribed)
RTI_EXCEPT(ObjectNotKnown)
RTI_EXCEPT(ObjectAlreadyRegistered)
RTI_EXCEPT(RegionNotKnown)
RTI_EXCEPT(RestoreNotRequested)
RTI_EXCEPT(RTIinternalError)
RTI_EXCEPT(SpaceNotDefined)
RTI_EXCEPT(SaveInProgress)
RTI_EXCEPT(SaveNotInitiated)
RTI_EXCEPT(SpecifiedSaveLabelDoesNotExist)
RTI_EXCEPT(TimeAdvanceAlreadyInProgress)
RTI_EXCEPT(TimeAdvanceWasNotInProgress)
RTI_EXCEPT(TooManyIDsRequested)
RTI_EXCEPT(UnableToPerformSave)
RTI_EXCEPT(UnimplementedService)
RTI_EXCEPT(UnknownLabel)
RTI_EXCEPT(ValueCountExceeded)
RTI_EXCEPT(ValueLengthExceeded)

enum OrderType {
    RECEIVE = 1,
    TIMESTAMP
};

enum OwnershipDivestitureCondition {
    NEGOTIATED = 1,
    UNCONDITIONAL
};

enum ObjectRemovalReason {
    OUT_OF_REGION = 1,
    OBJECT_DELETED,
    NO_LONGER_SUBSCRIBED
};

```

```

enum TransportType {
    RELIABLE = 1,
    BEST_EFFORT
};

enum ResignAction {
    RELEASE_ATTRIBUTES = 1,
    DELETE_OBJECTS,
    DELETE_OBJECTS_AND_RELEASE_ATTRIBUTES,
    NO_ACTION
};

class FederateAmbassador;

typedef FederateAmbassador *FederateAmbassadorPtr;

typedef RTI::UShort ObjectClassHandle;

typedef RTI::UShort InteractionClassHandle;

typedef RTI::UShort Handle;

typedef Handle AttributeHandle;

typedef Handle ParameterHandle;

typedef RTI::UShort FederateHandle;

typedef RTI::ULong FederateID;

typedef RTI::ULong ObjectID;

typedef RTI::ULong UniqueID;

class HandleValuePairSet {
public:
    virtual ~HandleValuePairSet() { ; }

    virtual ULong      // returned C3
    size() const = 0; // Cardinality of the set.

    virtual Handle // returned C3
    getHandle(RTI::ULong i) const
    throw (
        ArrayIndexOutOfBoundsException) = 0;

    virtual void getValue(RTI::ULong i,
        char *buff,           // supplied C4
        ULong &valueLength) const // returned C2
    throw (
        ArrayIndexOutOfBoundsException) = 0;

    virtual void add(     // Add new value.
        Handle h,           // supplied C1
        const char *buff,   // supplied C4
        ULong valueLength) // supplied C1
    throw (
        ValueLengthExceeded,
        ValueCountExceeded) = 0;

    virtual void moveFrom(const HandleValuePairSet& ahvp,
                           RTI::ULong &i) // supplied C1
    throw (

```

```

ValueCountExceeded,
ArrayIndexOutOfBoundsException) = 0;

virtual void empty() = 0; // Empty the Set without deallocating space.

virtual inline RTI::ULong start() const = 0;
virtual inline RTI::ULong valid(RTI::ULong i) const = 0;
virtual inline RTI::ULong next(RTI::ULong i) const = 0;
};

typedef HandleValuePairSet AttributeHandleValuePairSet;

class AttributeSetFactory {
public:
    static AttributeHandleValuePairSet // returned C6
    *create( // Create set with space for
              // specified number of attributes.
        ULong count) // supplied C1
    throw (
        MemoryExhausted,
        ValueCountExceeded);
};

class AttributeHandleSet {
public:
    AttributeHandleSet();
    ULong // returned C3
    size() const; // Cardinality of the set.

    AttributeHandle // returned C3
    getHandle( // Get ith handle.
        ULong i) const // supplied C1
    throw (
        ArrayIndexOutOfBoundsException);

    void add( // Add new value.
        AttributeHandle h) // supplied C1
    throw (
        AttributeNotDefined);

    void empty(); // Empty the Set

    Boolean isEmpty(); // is set empty?
    Boolean isMember(AttributeHandle h) const;
    AttributeHandleSet setUnion(AttributeHandleSet &);
    AttributeHandleSet setIntersection(AttributeHandleSet &);
    AttributeHandleSet removeSetIntersection(AttributeHandleSet &);

    ULong encodedLength() const;
    void encode(char *buff) const;
    static AttributeHandleSet *decode(const char *buff)
        throw(ValueCountExceeded);

    friend ostream& operator << (ostream&, AttributeHandleSet &);

private:
    unsigned short _size;
    unsigned long _words[(MAX_ATTRIBUTES_PER_CLASS-1)/32+1];
};

class AttributeHandleSetFactory {
public:
    static AttributeHandleSet // returned C6

```

```

*create(                                // Create set with space for specified
                                         // number of AttributeHandles.
    ULong count)                      // supplied C1
throw(
    MemoryExhausted,
    ValueCountExceeded);
};

typedef RTI::UShort ObjectIDcount;

typedef RTI::Double FederationTimeDelta;

typedef RTI::Double FederationTime;

typedef RTI::Double TickTime;

class FederateHandleSet {
public:
    virtual ULong      // returned C3
    size() const = 0; // Cardinality of the set.

    virtual FederateHandle // returned C3
    getHandle(           // Get ith handle.
        ULong i) const   // supplied C1
    throw (
        ArrayIndexOutOfBoundsException) = 0;

    virtual void add(    // Add new value.
        FederateHandle h) // supplied C1
    throw (
        ValueCountExceeded) = 0;

    virtual void empty() = 0; // Empty the set without deallocating space.
};

class FederateHandleSetFactory {
public:
    static FederateHandleSet // returned C6
    *create(                // Create set with space for specified
                           // number of FederateHandles.
        ULong count)        // supplied C1
    throw (
        MemoryExhausted,
        ValueCountExceeded);
};

struct EventRetractionHandle_s {
    FederationTime theEventTime;
    UniqueID theSerialNumber;
    FederateHandle sendingFederate;
};
typedef struct EventRetractionHandle_s EventRetractionHandle;

// All char * declarations should use null terminated strings

typedef char * FederationExecutionName;

typedef char * FederateName;

typedef char * FileName;

typedef char * PauseLabel;

```

```

typedef char * SaveLabel;

typedef char * UserSuppliedTag;

typedef char * ObjectClassName;

typedef char * AttributeName;

typedef char * InteractionClassName;

typedef char * ParameterName;

typedef HandleValuePairSet ParameterHandleValuePairSet;

class ParameterSetFactory {
public:
    static ParameterHandleValuePairSet // returned C6
    *create( // Create set with space for
              // specified number of attributes.
        ULong count) // supplied C1
    throw (
        MemoryExhausted,
        ValueCountExceeded);
};

struct range_struct_s {
    RTI::ULong first;
    RTI::ULong last;
};
typedef struct range_struct_s Range;

struct Extents_s {
    RTI::ULong _length;
    Range* _buffer;
};
typedef struct Extents_s Extents;

struct ExtentSet_s {
    RTI::ULong _length;
    Extents* _buffer;
};
typedef struct ExtentSet_s ExtentSet;

typedef RTI::ULong Region;

struct Threshold_s {
    RTI::ULong value;
};
typedef Threshold_s Threshold;

struct ThresholdSet_s {
    RTI::ULong _length;
    Threshold* _buffer;
};
typedef struct ThresholdSet_s ThresholdSet;

typedef RTI::Long SpaceHandle;

typedef char * SpaceName;

```

```

//File: RTIambServices.hh

//          RTI Parameter Passing Memory Conventions
//         

// C1  In parameter by value.
// C2  Out parameter by reference.
// C3  Function return by value.
// C4  In parameter by const reference.  Caller provides memory.
//      Caller may free memory or overwrite it upon completion of
//      the call.  Callee must copy during the call anything it
//      wishes to save beyond completion of the call.  Parameter
//      type must define const accessor methods.
// C5  Out parameter by reference.  Caller provides reference to object.
//      Callee constructs an instance on the heap (new) and returns.
//      The caller destroys the instance (delete) at its leisure.
// C6  Function return by reference.  Callee constructs an instance on
//      the heap (new) and returns a reference.  The caller destroys the
//      instance (delete) at its leisure.

typedef FederateAmbassador *FederateAmbassadorPtr;

///////////////////////////////
// Federation Management Services //
///////////////////////////////

// 2.1
void createFederationExecution (
    const FederationExecutionName executionName) // supplied C4
throw (
    FederationExecutionAlreadyExists,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 2.2
void destroyFederationExecution (
    const FederationExecutionName executionName) // supplied C4
throw (
    FederatesCurrentlyJoined,
    FederationExecutionDoesNotExist,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 2.3
FederateHandle
joinFederationExecution (                                // returned C3
    const FederateName           yourName,                // supplied C4
    const FederationExecutionName executionName,          // supplied C4
    FederateAmbassadorPtr       federateAmbassadorReference) // supplied C1
throw (
    FederateAlreadyExecutionMember,
    FederationExecutionDoesNotExist,
    CouldNotOpenFED,
    ErrorReadingFED,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 2.4
void resignFederationExecution (
    ResignAction theAction) // supplied C1
throw (
    FederateOwnsAttributes,
    FederateNotExecutionMember,
    FederationExecutionDoesNotExist,

```

```

    InvalidResignAction,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 2.5
void requestPause (      // not implemented in F.0
    const PauseLabel label) // supplied C4
throw (
    FederationAlreadyPaused,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 2.7
void pauseAchieved (      // not implemented in F.0
    const PauseLabel label) // supplied C4
throw (
    UnknownLabel,
    NoPauseRequested,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 2.8
void requestResume () // not implemented in F.0
throw (
    FederationNotPaused,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 2.10
void resumeAchieved () // not implemented in F.0
throw (
    NoResumeRequested,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 2.11
void requestFederationSave ( // not implemented in F.0
    const SaveLabel label, // supplied C4
    FederationTime theTime) // supplied C1
throw (
    FederationTimeAlreadyPassed,
    InvalidFederationTime,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

void requestFederationSave ( // not implemented in F.0
    const SaveLabel label) // supplied C4
throw (
    SaveInProgress,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

```

```

// 2.13
void federateSaveBegun () // not implemented in F.0
throw (
    SaveNotInitiated,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

void federateSaveBegun ( // not implemented in F.0
    FederationTime theTime) // supplied C1
throw (
    SaveNotInitiated,
    InvalidFederationTime,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 2.14
void federateSaveAchieved () // not implemented in F.0
throw (
    SaveNotInitiated,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

void federateSaveNotAchieved () // not implemented in F.0
throw (
    SaveNotInitiated,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 2.15
void requestRestore ( // not implemented in F.0
    const SaveLabel label) // supplied C4
throw (
    SpecifiedSaveLabelDoesNotExist,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 2.17
void restoreAchieved ( // not implemented in F.0
    const SaveLabel label) // supplied C4
throw (
    UnknownLabel,
    RestoreNotRequested,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

void restoreNotAchieved ( // not implemented in F.0
    const SaveLabel label) // supplied C4
throw (
    UnknownLabel,
    RestoreNotRequested,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,

```

```

RTIinternalError,
UnimplementedService);

///////////////////////////////
// Declaration Management Services //
///////////////////////////////

// 3.1
void publishObjectClass (
    ObjectClassHandle theClass,           // supplied C1
    const AttributeHandleSet& attributeList) // supplied C4
throw (
    ObjectClassNotDefined,
    AttributeNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

void unpublishObjectClass (
    ObjectClassHandle theClass) // supplied C1
throw (
    ObjectClassNotDefined,
    FederateOwnsAttributes,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 3.2
void publishInteractionClass (
    InteractionClassHandle theInteraction) // supplied C1
throw (
    InteractionClassNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

void unpublishInteractionClass (
    InteractionClassHandle theInteraction) // supplied C1
throw (
    InteractionClassNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 3.3
void subscribeObjectClassAttribute (
    ObjectClassHandle theClass,           // supplied C1
    const AttributeHandleSet& attributeList) // supplied C4
throw (
    ObjectClassNotDefined,
    AttributeNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

void subscribeObjectClassAttribute ( // not implemented in F.0
    ObjectClassHandle theClass,           // supplied
    AttributeHandle theAttribute,        // supplied
    Region theRegion)                  // supplied
throw (
    ObjectClassNotDefined,
    AttributeNotDefined,
    RegionNotKnown,
    FederateNotExecutionMember,

```



```

///////////////////////////////
// 4.1
void requestID (
    ObjectIDcount idCount, // supplied C1
    ObjectID& firstID, // returned C2
    ObjectID& lastID) // returned C2
throw (
    TooManyIDsRequested,
    IDsupplyExhausted,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.2
void registerObject (
    ObjectClassHandle theClass, // supplied C1
    ObjectID theObject) // supplied C1
throw (
    InvalidObjectID,
    ObjectAlreadyRegistered,
    ObjectClassNotDefined,
    ObjectClassNotPublished,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.3
EventRetractionHandle updateAttributeValues ( // returned C3
    ObjectID theObject, // supplied C1
    const AttributeHandleValuePairSet& theAttributes, // supplied C4
    FederationTime theTime, // supplied C1
    const UserSuppliedTag theTag) // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotDefined,
    AttributeNotOwned,
    InvalidFederationTime,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.6
EventRetractionHandle sendInteraction ( // returned C3
    InteractionClassHandle theInteraction, // supplied C1
    const ParameterHandleValuePairSet& theParameters, // supplied C4
    FederationTime theTime, // supplied C1
    const UserSuppliedTag theTag) // supplied C4
throw (
    InteractionClassNotPublished,
    InteractionClassNotDefined,
    InteractionParameterNotDefined,
    InvalidFederationTime,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.8
EventRetractionHandle deleteObject ( // returned C3
    ObjectID objectID, // supplied C1
    FederationTime theTime, // supplied C1

```

```

    const UserSuppliedTag theTag) // supplied C4
throw (
    DeletePrivilegeNotHeld,
    ObjectNotKnown,
    InvalidFederationTime,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.10
void changeAttributeTransportType (
    ObjectId          theObject,      // supplied C1
    const AttributeHandleSet& theAttributes, // supplied C4
    TransportType       theType)        // supplied C1
throw (
    ObjectNotKnown,
    AttributeNotDefined,
    AttributeNotOwned,
    InvalidTransportType,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.11
void changeAttributeOrderType (
    ObjectId          theObject,      // supplied C1
    const AttributeHandleSet& theAttributes, // supplied C4
    OrderType         theType)        // supplied C1
throw (
    ObjectNotKnown,
    AttributeNotDefined,
    AttributeNotOwned,
    InvalidOrderType,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.12
void changeInteractionTransportType (
    InteractionClassHandle theClass, // supplied C1
    TransportType          theType) // supplied C1
throw (
    InteractionClassNotDefined,
    InteractionClassNotPublished,
    InvalidTransportType,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.13
void changeInteractionOrderType (
    InteractionClassHandle theClass, // supplied C1
    OrderType              theType) // supplied C1
throw (
    InteractionClassNotDefined,
    InteractionClassNotPublished,
    InvalidOrderType,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.14
void requestObjectAttributeValueUpdate (
    ObjectId          theObject,      // supplied C1

```

```

    const AttributeHandleSet& theAttributes) // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

void requestObjectAttributeValueUpdate (
    ObjectID theObject) // supplied C1
throw (
    ObjectNotKnown,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

void requestClassAttributeValueUpdate (
    ObjectClassHandle theClass,           // supplied C1
    const AttributeHandleSet& theAttributes) // supplied C4
throw (
    ObjectClassNotDefined,
    AttributeNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

void requestClassAttributeValueUpdate (
    ObjectClassHandle theClass) // supplied C1
throw (
    ObjectClassNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 4.16
void retract (
    EventRetractionHandle theHandle) // supplied C1
throw (
    InvalidRetractionHandle,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

///////////////////////////////
// Ownership Management Services //
///////////////////////////////

// 5.1
void requestAttributeOwnershipDivestiture (
    ObjectID                  theObject,      // supplied C1
    const AttributeHandleSet& theAttributes, // supplied C4
    OwnershipDivestitureCondition theCondition, // supplied C1
    const UserSuppliedTag       theTag)        // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotDefined,
    AttributeNotOwned,
    InvalidDivestitureCondition,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

void requestAttributeOwnershipDivestiture (
    ObjectID                  theObject,      // supplied C1

```

```

const AttributeHandleSet&           theAttributes, // supplied C4
                                      OwnershipDivestitureCondition theCondition, // supplied C1
const UserSuppliedTag                theTag,        // supplied C4
const FederateHandleSet&            theCandidates) // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotDefined,
    AttributeNotOwned,
    InvalidDivestitureCondition,
    FederateDoesNotExist,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 5.5
void requestAttributeOwnershipAcquisition (
    ObjectId             theObject,          // supplied C1
    const AttributeHandleSet& desiredAttributes, // supplied C4
    const UserSuppliedTag      theTag)         // supplied C4
throw (
    ObjectNotKnown,
    ObjectClassNotPublished,
    ObjectClassNotSubscribed,
    AttributeNotDefined,
    AttributeNotPublished,
    AttributeNotSubscribed,
    FederateOwnsAttributes,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 5.7
void queryAttributeOwnership (
    ObjectId             theObject,          // supplied C1
    AttributeHandle theAttribute) // supplied C1
throw (
    ObjectNotKnown,
    AttributeNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 5.9
RTI::Boolean                      // returned C3
attributeIsOwnedByFederate (
    ObjectId             theObject,          // supplied C1
    AttributeHandle theAttribute) // supplied C1
throw (
    ObjectNotKnown,
    AttributeNotDefined,
    AttributeNotKnown,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

///////////////////////////////
// Time Management Services //
///////////////////////////////

// 6.1
FederationTime // returned C3
requestFederationTime ()
throw (
    FederateNotExecutionMember,

```

```

ConcurrentAccessAttempted,
RTIinternalError);

// 6.2
FederationTime // returned C3
requestLBTS ()
throw (
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 6.3
FederationTime // returned C3
requestFederateTime ()
throw (
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 6.4
FederationTime // returned C3
requestMinNextEventTime ()
throw (
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 6.5
void setLookahead (
    FederationTimeDelta theLookahead) // supplied C1
throw (
    InvalidLookahead,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 6.6
FederationTimeDelta // returned C3
requestLookahead ()
throw (
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 6.7
void timeAdvanceRequest (
    FederationTime theTime) // supplied C1
throw (
    InvalidFederationTime,
    TimeAdvanceAlreadyInProgress,
    FederationTimeAlreadyPassed,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 6.8
void nextEventRequest (
    FederationTime theTime) // supplied C1
throw (
    TimeAdvanceAlreadyInProgress,
    FederationTimeAlreadyPassed,
    InvalidFederationTime,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,

```

```

RTIinternalError);

// 6.9
void flushQueueRequest (
    FederationTime theTime) // supplied C1
throw (
    InvalidFederationTime,
    TimeAdvanceAlreadyInProgress,
    FederationTimeAlreadyPassed,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

///////////////////////////////
// Data Distribution Management //
///////////////////////////////

// 7.1
void createUpdateRegion ( // not implemented in F.0
    SpaceHandle theSpace, // supplied
    ExtentSet& theExtents, // supplied
    Region& theRegion) // returned
throw (
    SpaceNotDefined,
    InvalidExtents,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 7.2
void createSubscriptionRegion ( // not implemented in F.0
    SpaceName theSpace, // supplied
    ExtentSet& theExtents, // supplied
    Region& theRegion) // returned
throw (
    SpaceNotDefined,
    InvalidExtents,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 7.3
void associateUpdateRegion ( // not implemented in F.0
    Region theRegion, // supplied
    ObjectID theObject, // supplied
    AttributeHandleSet& theAttributes) // supplied
throw (
    RegionNotKnown,
    ObjectNotKnown,
    AttributeNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

void disassociateUpdateRegion ( // not implemented in F.0
    Region theRegion, // supplied
    ObjectID theObject, // supplied
    AttributeHandleSet& theAttributes) // supplied
throw (
    RegionNotKnown,
    ObjectNotKnown,

```

```

AttributeNotDefined,
FederateNotExecutionMember,
ConcurrentAccessAttempted,
RTIinternalError,
UnimplementedService);

void associateUpdateRegion (      // not implemented in F.0
    Region          theRegion, // supplied
    InteractionClassHandle theClass) // supplied
throw (
    RegionNotKnown,
    InteractionClassNotKnown,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

void disassociateUpdateRegion (      // not implemented in F.0
    Region          theRegion, // supplied
    InteractionClassHandle theClass) // supplied
throw (
    RegionNotKnown,
    InteractionClassNotKnown,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 7.5
void modifyRegion (      // not implemented in F.0
    Region          theRegion, // supplied
    ExtentSet& theExtents) // supplied
throw (
    RegionNotKnown,
    InvalidExtents,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 7.6
void deleteRegion ( // not implemented in F.0
    Region theRegion) // supplied
throw (
    RegionNotKnown,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

///////////////////////////////
// RTI Support Services //
////////////////////////////

// 8.1
ObjectClassHandle           // returned C3
getObjectClassHandle (
    const ObjectClassName theName) // supplied C4
throw (
    NameNotFound,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

```

```

// 8.2
ObjectClassName           // returned C6
getObjectName (
    ObjectClassHandle theHandle) // supplied C1
throw (
    ObjectClassNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 8.3
AttributeHandle           // returned C3
getAttributeHandle (
    const AttributeName theName,      // supplied C4
    ObjectClassHandle whichClass) // supplied C1
throw (
    ObjectClassNotDefined,
    NameNotFound,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 8.4
AttributeName             // returned C6
getAttributeName (
    AttributeHandle theHandle, // supplied C1
    ObjectClassHandle whichClass) // supplied C1
throw (
    ObjectClassNotDefined,
    AttributeNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 8.5
InteractionClassHandle    // returned C1
getInteractionClassHandle (
    const InteractionClassName theName) // supplied C4
throw (
    NameNotFound,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 8.6
InteractionClassName       // returned C6
getInteractionClassName (
    InteractionClassHandle theHandle) // supplied C1
throw (
    InteractionClassNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 8.7
ParameterHandle            // returned C1
getParameterHandle (
    const ParameterName theName,      // supplied C4
    InteractionClassHandle whichClass) // supplied C1
throw (
    InteractionClassNotDefined,
    NameNotFound,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

```

```

RTIinternalError);

// 8.8
ParameterName           // returned C6
getParameterName (
    ParameterHandle      theHandle, // supplied C1
    InteractionClassHandle whichClass) // supplied C1
throw (
    InteractionClassNotDefined,
    InteractionParameterNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

// 8.9
SpaceHandle           // returned C1
getSpaceHandle (        // not implemented in F.0
    const SpaceName theName) // supplied C4
throw (
    NameNotFound,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 8.10
SpaceName             // returned C6
getSpaceName (          // not implemented in F.0
    const SpaceHandle theHandle) // supplied C4
throw (
    SpaceNotDefined,
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError,
    UnimplementedService);

// 8.11
void RTI::RTIambassador::turnRegulationOn ()
throw (
    RTI::FederationTimeAlreadyPassed,
    RTI::FederateNotExecutionMember,
    RTI::ConcurrentAccessAttempted,
    RTI::RTIinternalError);

RTI::FederationTime           // Returned C3
RTI::RTIambassador::turnRegulationOnNow ()
throw (
    RTI::FederateNotExecutionMember,
    RTI::ConcurrentAccessAttempted,
    RTI::RTIinternalError);

void RTI::RTIambassador::turnRegulationOff ()
throw (
    RTI::FederateNotExecutionMember,
    RTI::ConcurrentAccessAttempted,
    RTI::RTIinternalError);

// 8.12
void setTimeConstrained (
    RTI::Boolean state) // supplied C1
throw (
    FederateNotExecutionMember,
    ConcurrentAccessAttempted,
    RTIinternalError);

```

```
// 8.13
RTI::Boolean // returned C3
tick ()
throw (
    ConcurrentAccessAttempted,
    RTIinternalError);

RTI::Boolean      // returned C3
tick (
    TickTime minimum, // supplied C1
    TickTime maximum) // supplied C1
throw (
    ConcurrentAccessAttempted,
    RTIinternalError);

RTIambassador()
throw (
    MemoryExhausted,
    RTIinternalError);

~RTIambassador()
throw (RTIinternalError);
```

```

//File: federateAmbServices.hh

//
// RTI Parameter Passing Memory Conventions
//

// C1 In parameter by value.
// C2 Out parameter by reference.
// C3 Function return by value.
// C4 In parameter by const reference. Caller provides memory.
// Caller may free memory or overwrite it upon completion of
// the call. Callee must copy during the call anything it
// wishes to save beyond completion of the call. Parameter
// type must define const accessor methods.
// C5 Out parameter by reference. Caller provides reference to object.
// Callee constructs an instance on the heap (new) and returns.
// The caller destroys the instance (delete) at its leisure.
// C6 Function return by reference. Callee constructs an instance on
// the heap (new) and returns a reference. The caller destroys the
// instance (delete) at its leisure.

///////////////////////////////
// Federation Management Services //
///////////////////////////////

// 2.6
virtual void initiatePause (
    const PauseLabel label) // supplied C4
throw (
    FederateAlreadyPaused,
    FederateInternalError) = 0;

// 2.9
virtual void initiateResume ()
throw (
    FederateNotPaused,
    FederateInternalError) = 0;

// 2.12
virtual void initiateFederateSave (
    const SaveLabel label) // supplied C4
throw (
    UnableToPerformSave,
    FederateInternalError) = 0;

virtual void initiateFederateSave (
    const SaveLabel label, // supplied C4
    FederationTime theTime) // supplied C1
throw (
    InvalidFederationTime,
    UnableToPerformSave,
    FederateInternalError) = 0;

// 2.16
virtual void initiateRestore (
    const SaveLabel label) // supplied C4
throw (
    SpecifiedSaveLabelDoesNotExist,
    CouldNotRestore,
    FederateInternalError) = 0;

///////////////////////////////
// Declaration Management Services //
/////////////////////////////

```

```

// 3.5
virtual void startUpdates (
    ObjectClassHandle    theClass,      // supplied C1
    const AttributeHandleSet& theAttributes) // supplied C4
throw (
    ObjectClassNotPublished,
    AttributeNotPublished,
    FederateInternalError) = 0;

virtual void stopUpdates (
    ObjectClassHandle    theClass,      // supplied C1
    const AttributeHandleSet& theAttributes) // supplied C4
throw (
    ObjectClassNotPublished,
    AttributeNotPublished,
    FederateInternalError) = 0;

// 3.6
virtual void startInteractionGeneration (
    InteractionClassHandle theHandle) // supplied C1
throw (
    InteractionClassNotPublished,
    FederateInternalError) = 0;

virtual void stopInteractionGeneration (
    InteractionClassHandle theHandle) // supplied C1
throw (
    InteractionClassNotPublished,
    FederateInternalError) = 0;

///////////////////////////////
// Object Management Services //
///////////////////////////////

// 4.4
virtual void discoverObject (
    ObjectId             theObject,      // supplied C1
    ObjectClassHandle    theObjectClass, // supplied C1
    FederationTime       theTime,        // supplied C1
    const UserSuppliedTag theTag,        // supplied C4
    EventRetractionHandle theHandle)     // supplied C1
throw (
    CouldNotDiscover,
    ObjectClassNotKnown,
    InvalidFederationTime,
    FederateInternalError) = 0;

// 4.5
virtual void reflectAttributeValue (
    ObjectId             theObject,      // supplied C1
    const AttributeHandleValuePairSet& theAttributes, // supplied C4
    FederationTime       theTime,        // supplied C1
    const UserSuppliedTag theTag,        // supplied C4
    EventRetractionHandle theHandle)     // supplied C1
throw (
    ObjectNotKnown,
    AttributeNotKnown,
    InvalidFederationTime,
    FederateInternalError) = 0;

// 4.7
virtual void receiveInteraction (
    InteractionClassHandle    theInteraction, // supplied C1
    const ParameterHandleValuePairSet& theParameters, // supplied C4

```

```

        FederationTime          theTime,           // supplied C1
const UserSuppliedTag      theTag,            // supplied C4
        EventRetractionHandle   theHandle)        // supplied C1
throw (
    InteractionClassNotKnown,
    InteractionParameterNotKnown,
    InvalidFederationTime,
    FederateInternalError) = 0;

// 4.9
virtual void removeObject (
    ObjectID                 theObject, // supplied C1
    ObjectRemovalReason      theReason, // supplied C1
    FederationTime           theTime,   // supplied C1
    const UserSuppliedTag    theTag,    // supplied C4
    EventRetractionHandle    theHandle) // supplied C1
throw (
    ObjectNotKnown,
    InvalidFederationTime,
    FederateInternalError) = 0;

virtual void removeObject (
    ObjectID                 theObject, // supplied C1
    ObjectRemovalReason      theReason) // supplied C1
    FederationTime           theTime,   // supplied C1
    const UserSuppliedTag    theTag)    // supplied C4
throw (
    ObjectNotKnown,
    InvalidFederationTime,
    FederateInternalError) = 0;

// 4.15
virtual void provideAttributeValueUpdate (
    ObjectID                 theObject, // supplied C1
    const AttributeHandleSet& theAttributes) // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotKnown,
    FederateInternalError) = 0;

// 4.17
virtual void reflectRetraction (
    EventRetractionHandle theHandle) // supplied C1
throw (
    EventNotKnown,
    FederateInternalError) = 0;

///////////////////////////////
// Ownership Management Services //
///////////////////////////////

// 5.2
virtual AttributeHandleSet&                                // returned C6
requestAttributeOwnershipAssumption (
    ObjectID                 theObject,           // supplied C1
    const AttributeHandleSet& offeredAttributes, // supplied C4
    const UserSuppliedTag    theTag)            // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotKnown,
    AttributeAlreadyOwned,
    FederateInternalError) = 0;

// 5.3

```

```

virtual void attributeOwnershipDivestitureNotification (
    ObjectID             theObject,           // supplied C1
    const AttributeHandleSet& releasedAttributes) // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotKnown,
    FederateInternalError) = 0;

// 5.4
virtual void attributeOwnershipAcquisitionNotification (
    ObjectID             theObject,           // supplied C1
    const AttributeHandleSet& securedAttributes) // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotKnown,
    FederateInternalError) = 0;

// 5.6
virtual AttributeHandleSet&                                // returned C6
requestAttributeOwnershipRelease (
    ObjectID             theObject,           // supplied C1
    const AttributeHandleSet& candidateAttributes, // supplied C4
    const UserSuppliedTag      theTag)           // supplied C4
throw (
    ObjectNotKnown,
    AttributeNotKnown,
    FederateInternalError) = 0;

// 5.8
virtual void informAttributeOwnership (
    ObjectID             theObject,           // supplied C1
    AttributeHandle theAttribute, // supplied C1
    FederateHandle theOwner)      // supplied C1
throw (
    ObjectNotKnown,
    AttributeNotKnown,
    FederateInternalError) = 0;

///////////////////////////////
// Time Management Services //
///////////////////////////////

// 6.10
virtual void timeAdvanceGrant (
    FederationTime theTime) // supplied C1
throw (
    InvalidFederationTime,
    TimeAdvanceWasNotInProgress,
    FederationTimeAlreadyPassed,
    FederateInternalError) = 0;

/////////////////////////////
// Data Distribution Management //
/////////////////////////////

// 7.4
virtual void changeThresholds (
    Region             theRegion,           // supplied
    ThresholdSet& theThresholds) // returned
throw (
    RegionNotKnown,
    FederateInternalError) = 0;

```

```
//File: RTI.hh

#ifndef RTI_hh
#define RTI_hh

#include <fstream.h>
#include <math.h>

struct RTIambPrivate;

class RTI {
public:

    #include "baseTypes.hh"
    #include "RTItypes.hh"

    class RTIambassador {
public:
    #include "RTIambServices.hh"
private:
    RTIambPrivate* privateData;
};

    class FederateAmbassador {
public:
    #include "federateAmbServices.hh"
};
};

#endif
```